

# Esame di Calcolatori Elettronici T

## 18 Gennaio 2018 (Ing. Informatica)

### Esercizio 1

In un sistema basato sul DLX, con **1 GB di EPROM** mappata negli indirizzi bassi e **1 GB di RAM** mappata negli indirizzi alti, sono presenti **quattro porte a 8 bit in INPUT (IN\_0, IN\_1, IN\_2 e IN\_3)**, già progettate, basate sul protocollo di *handshake*. Si desidera progettare un sistema che consenta di gestire, **in modo completamente automatico**, la seguente **sequenza di operazioni** sulle 4 porte:

- 3 letture consecutive da IN\_0
- 4 letture consecutive da IN\_1
- 2 letture consecutive da IN\_2
- 1 lettura da IN\_3
- e così via (i.e., 3 letture consecutive da IN\_0, ecc)

Per ogni ciclo di lettura appena indicato, i dati letti dalle porte in input dovranno essere scritti nei seguenti indirizzi di memoria:

- dati letti da IN\_0 scritti da **FFFF0000h** (primo) a **FFFF0002h** (ultimo)
- dati letti da IN\_1 scritti da **FFFF0004h** (primo) a **FFFF0007h** (ultimo)
- dati letti da IN\_2 scritti da **FFFF0008h** (primo) a **FFFF0009h** (ultimo)
- dato letto da IN\_3 scritto a **FFFF000Ch** (unico)

All'avvio del sistema debbono essere abilitate le **tre letture da IN\_0**.

- **Descrivere sinteticamente la soluzione** che s'intende realizzare e indicare **chiaramente quali sono i segnali di *chip-select* necessari**
- Progettare il sistema **minimizzando le risorse necessarie evidenziando e risolvendo eventuali criticità**
- Indicare le espressioni di decodifica e il range d'indirizzi di tutte le periferiche, le memorie e i segnali utilizzati
- Scrivere il codice **ottimizzato** dell'*interrupt handler* (i registri da R25 a R29 possono essere utilizzati senza la necessità di doverli ripristinare)
- Soluzioni **interamente software e NON saranno considerate valide**

### Esercizio 2

Indicare quali potrebbero essere (tutti) i segnali d'ingresso e di uscita della rete logica Register File utilizzato dal DLX. Per ogni segnale si aggiunga una breve descrizione.

### Esercizio 3

Indicare sinteticamente quali sono i vantaggi principali derivanti dalla codifica delle istruzioni adottata dal DLX.

## MAPPING

1 GB EPROM		1GB RAM
(512+512)		

EPROM :  $0 \times 00000000 \rightarrow 0 \times 3FFFFFFF$  ( $4 \times 256 \text{ MB}$ )

RAM\_512\_L :  $0 \times C0000000 \rightarrow 0 \times DFFFFFFF$  ( $4 \times 128 \text{ MB}$ )

RAM\_512\_H :  $0 \times E0000000 \rightarrow 0 \times FFFFFFFF$  ( $4 \times 128 \text{ MB}$ )

## CHIP SELECT

CS\_RAM\_L\_0 = BA31  $\overline{\text{BA29}}$  BE 0

CS\_RAM\_L\_1 = BA31  $\overline{\text{BA29}}$  BE 1

CS\_RAM\_L\_2 = BA31  $\overline{\text{BA29}}$  BE 2

CS\_RAM\_L\_3 = BA31  $\overline{\text{BA29}}$  BE 3

CS\_RAM\_L\_0 = BA31 BA29 BE 0

CS\_RAM\_L\_1 = BA31 BA29 BE 1

CS\_RAM\_L\_2 = BA31 BA29 BE 2

CS\_RAM\_L\_3 = BA31 BA29 BE 3

CS\_EPROM\_0 =  $\overline{\text{BA31}}$  BA30 BE 0

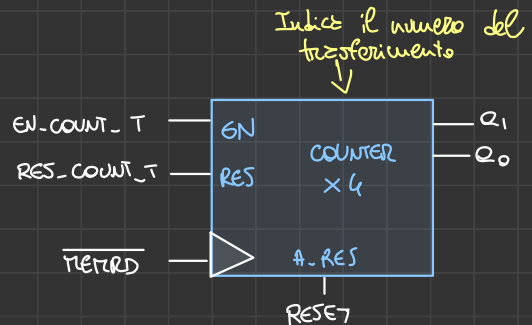
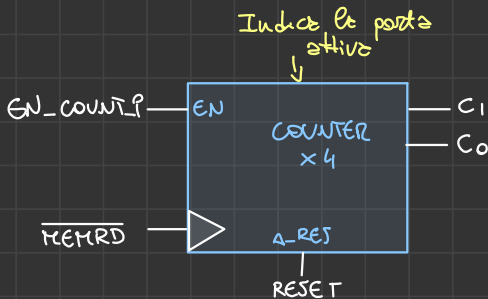
CS\_EPROM\_1 =  $\overline{\text{BA31}}$  BA30 BE 1

CS\_EPROM\_2 =  $\overline{\text{BA31}}$  BA30 BE 2

CS\_EPROM\_3 =  $\overline{\text{BA31}}$  BA30 BE 3

CS\_INPUT\_PORT =  $\overline{\text{BA31}}$  BA30 BE 0

CS\_READ\_ADDRESS = BA31 BA30 BE 1 (MEMRD)



$$\text{EN\_COUNT\_P} = \text{RES\_COUNT\_T}$$

$$\text{EN\_COUNT\_T} = \text{CS\_INPUT\_PORT}$$

$$\text{RES\_COUNT\_T} = \text{CS\_INPUT\_PORT} \cdot (1BF_0 \cdot \text{PORT}_0 \cdot Q_1 \overline{Q_0} + 1BF_1 \cdot \text{PORT}_1 \cdot Q_1 Q_0 + 1BF_2 \cdot \text{PORT}_2 \cdot \overline{Q_1} Q_0 + 1BF_3 \cdot \text{PORT}_3 \cdot \overline{Q_1} \overline{Q_0})$$

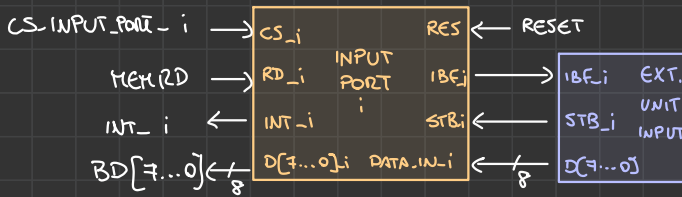
$$\text{PORT}_0 = \overline{C1} \cdot C0$$

$$\text{PORT}_1 = \overline{C1} \cdot \overline{C0}$$

$$\text{PORT}_2 = C1 \cdot \overline{C0}$$

$$\text{PORT}_3 = C1 \cdot C0$$

## PORT6 IN INPUT



$$CS\_INPUT\_PORT\_0 = CS\_INPUT\_PORT \cdot PORT\_0 \cdot IBF\_0$$

$$CS\_INPUT\_PORT\_1 = CS\_INPUT\_PORT \cdot PORT\_1 \cdot IBF\_1$$

$$CS\_INPUT\_PORT\_2 = CS\_INPUT\_PORT \cdot PORT\_2 \cdot IBF\_2$$

$$CS\_INPUT\_PORT\_3 = CS\_INPUT\_PORT \cdot PORT\_3 \cdot IBF\_3$$

$$INT (TO DLX) = INT\_0 \cdot PORT\_0 + INT\_1 \cdot PORT\_1 + INT\_2 \cdot PORT\_2 + INT\_3 \cdot PORT\_3$$

- dati letti da IN\_0 scritti da FFFF0000h (primo) a FFFF0002h (ultimo)
- dati letti da IN\_1 scritti da FFFF0004h (primo) a FFFF0007h (ultimo)
- dati letti da IN\_2 scritti da FFFF0008h (primo) a FFFF0009h (ultimo)
- dato letto da IN\_3 scritto a FFFF000Ch (unico)

PORT 0 :  $C_1 C_0 Q_1 Q_0 \rightarrow 0001 \rightarrow 0010$  (0 → 2)

Port 1 → TRANSFER

PORT 1 :  $0100 \rightarrow 0111$  (4 → 7)

PORT 2 :  $1000 \rightarrow 1001$  (8 → 9)

PORT 3 :  $1100 \rightarrow 1100$  (12 → C<sub>16</sub>)

CS-READ-ADDRESS

C<sub>1</sub> →  $\rightarrow$  BD3

CS-READ-ADDRESS

C<sub>0</sub> →  $\rightarrow$  BD2

CS-READ-ADDRESS

Q<sub>1</sub> →  $\rightarrow$  BD1

CS-READ-ADDRESS

Q<sub>0</sub> →  $\rightarrow$  BD0

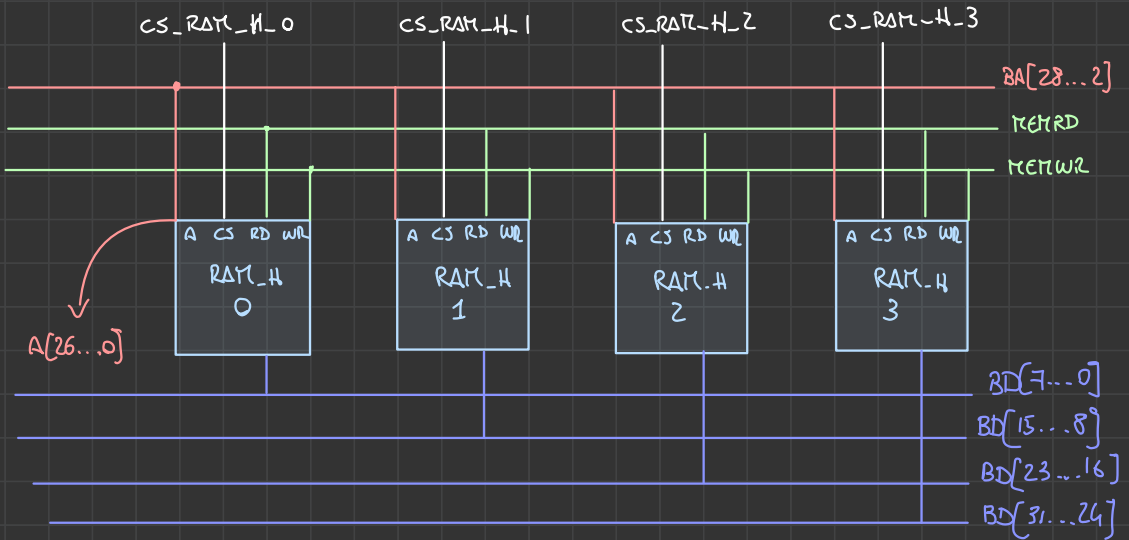
CS-READ-ADDRESS

(0)<sub>4</sub> →  $\rightarrow$  BD[7...4]

# CODICE

0h HANDLER: LHI R25, 0x4000; R25 = 0x40000000  
 4h LBU R26, 0x0001(R25); Leggo da CS\_READ\_ADDRESS (0x40000001)  
 8h LBU R27, 0x0000(R25); Leggo dato da CS\_INPUT\_PORT e lo registro in R27  
 ch  
 10h LHI R25, 0xFFFF; R25 = 0xFFFF0000  
 14h ADD R28, R25, R26; R28 = R25 + R26 = 0xFFFF0001 (C1C0Q1Q0)<sub>2</sub>  
 18h SB R27, 0x0000(R28) Registro contenuto di R27 in 0xFFFF0001  
 RFE;

## INTERFACCIAMENTO MEMORIE



$$512 \text{ MB} \rightarrow 512 \cdot 1024^2 = 2^9 \cdot 2^{20} = 2^{29}$$